DEVELOP PHASE

Rekonkati,Thupten T.K.

Introduction

In this document you can read about the process of the programming part. We tell what we have done **and** show a piece of code and the result. You can also read about other projects I have done such as protfolio and plexicastle website.

Table of Contents

Goal

Our goal is to make the app just like the final design using flutter, without errors or bugs so the users can have a positive experience.

Tools

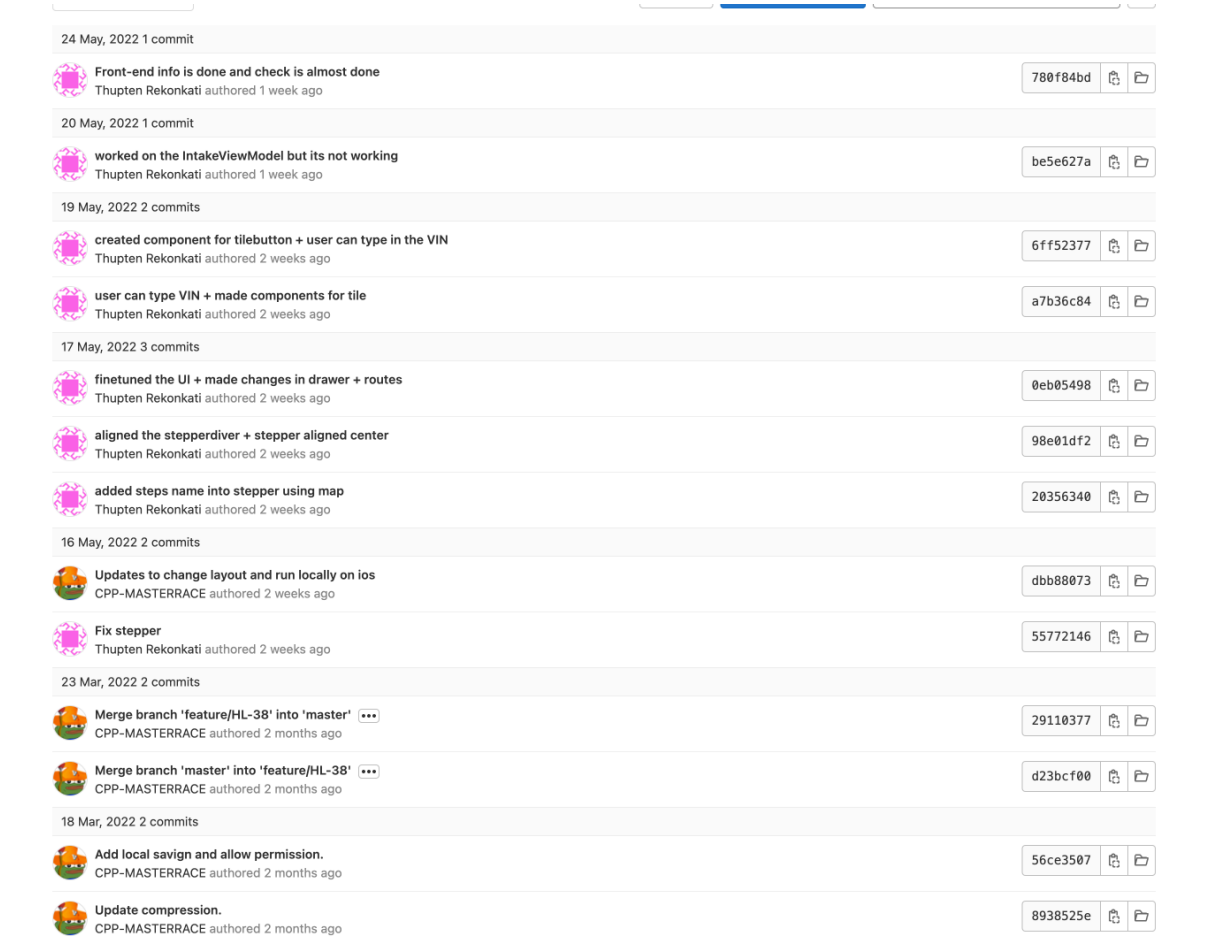
As we said in the define phase, we want to use Flutter. We chose this because the existing version is also made with flutter, we choose for a native app because it has better support for hardware. For our project we want to make use of camera and NFC. Furthermore, flutter app works on apple as well as on android.

For version control we used, GitLab. With the use of GitLab, GitHub Desktop and build-in VS code Git, we managed to work safely. Each time we got something to work we push it to the Git.

Git

We are not allowed to share the git link, but we added some part of the code in this document as a proof.

Screenshots of our commit



Intake and check app

We cloned the current intake and check app from GitLab and rebuild it, we choose to do this because we then could use some elements, like scanning barcode or QR-code.

Learning about global theme

At first, we had no knowledge regarding programming with flutter. So, we started to learn the basics and started to learn about global theme. We decided to learn this because we wanted to create the front-end, with global theme it's very easy to change a color by just adjusting a variable.

Code

```
return MaterialApp(
    scaffoldMessengerKey: snackbarKey,
    title: 'Hexite Software',
    theme: ThemeData(
      primaryColor: Color.fromARGB(255, 255, 145, 0),
      secondaryHeaderColor: Color.fromARGB(255, 235, 102, 45),
      backgroundColor: Color.fromARGB(255, 224, 224, 224),
      highlightColor: Colors.black,
      colorScheme:
          ColorScheme.fromSwatch(primarySwatch: Colors.deepOrange).copyWith(
        secondary: Color.fromARGB(255, 235, 100, 44),
      ),
    ), // ThemeData
```

Structure and layout

we didn't know much about flutter, so we then started learning about the structure and layout of flutter (Marshall, sd), how to build it up and soon we learned more about widgets and classes, that you had to build the code as a tree and had to stack widgets on top of each other. For us this was a breakthrough, because now we knew how to build up a page. We managed to build the home page with build in Top Bar with a menu. We also created the neomorphism button. This was not a normal button we had to create a container with styling and event listener.

Code

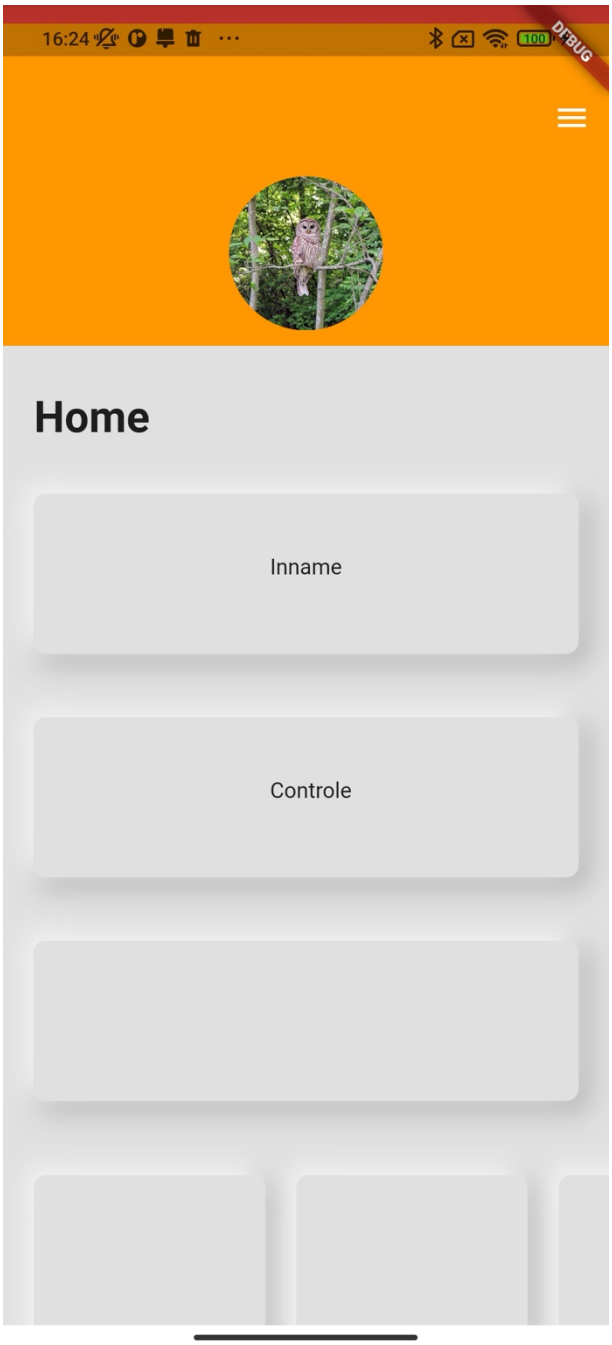
```dart
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';

import '../components/appbar.dart';
import '../view/drawer.dart';

You, 3 minutes ago | 3 authors (You and others)
class Home extends StatelessWidget {
  const Home({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    var size = MediaQuery.of(context).size;
    var tempState = "";
    return Scaffold(
      appBar: defaultAppBar(),
      endDrawer: DrawerView(),
      body: Container(
        width: size.width,
        decoration: BoxDecoration(color: Colors.grey[300]),
        child: Container(
          child: Padding(
            padding: const EdgeInsets.only(),
            child: Column(
              children: [
                Container(
                  decoration: BoxDecoration(color: Colors.orange),
                  child: Row(
                    children: <Widget>[
                      Expanded(
                        flex: 1,
                        child: Container(
                          height: 100,
                        ), // Container
                      ), // Expanded
                      Container(
                        height: 100,
                        width: 100,
                        margin: EdgeInsets.all(10),
                        child: ClipRRect(
                          borderRadius: BorderRadius.circular(1000),
                          child: Image(
                            image: NetworkImage(
                                'https://flutter.github.io/assets-for-api-docs/assets/widgets/owl-2.jpg'),
                          ), // Image
                        ), // ClipRRect
                      ), // Container
                      Expanded(
                        flex: 1,
                        child: Container(
                          height: 100,
                        ), // Container
                      ), // Expanded
                    ], // <Widget>[]
                  ), // Row
                ), // Container
                Row(
                  children: <Widget>[
                    Padding(
                      padding:
                          EdgeInsets.only(left: 20.0, top: 30.0, right: 200.0),
                      child: Container(
                        child: Row(
                          children: <Widget>[
                            Text(
                              "Home",
                              style: TextStyle(
                                  fontWeight: FontWeight.bold, fontSize: 28.0), // TextStyle
                            ), // Text
                          ], // <Widget>[]
                        ), // Row
                      ), // Container
                    ), // Padding
                  ], // <Widget>[]
                ), // Row
                Expanded(
                  child: ListView(
                    children: [
                      Container(
                        margin: EdgeInsets.only(top: 25.0),
                        width: double.infinity,
                        height: 120,
```

Figure 2 - home.dart

```dart
                        Container(
                          height: 150.0,
                          width:
                              MediaQuery.of(context).size.width * 0.9,
                          decoration: BoxDecoration(
                            borderRadius: BorderRadius.circular(8.0),
                            color: Colors.grey[300],
                            boxShadow: [
                              BoxShadow(
                                color: Colors.grey[500]!,
                                blurRadius: 15,
                                spreadRadius: 1,
                                offset: const Offset(4.0,          You, 3 weeks ago • Fix stepper
                                    4.0), // shadow direction: bottom right // Offset
                              ), // BoxShadow
                              BoxShadow(
                                color: Colors.white,
                                offset: Offset(-4.0, -4.0),
                                blurRadius: 15,
                                spreadRadius: 1,
                              ), // BoxShadow
                            ],
                          ), // BoxDecoration
                          child: Row(
                            children: [
                              Padding(
                                padding: EdgeInsets.only(left: 20)), // Padding
                              Container(
                                width: 100,
                                height: 25,
                                child: SvgPicture.asset(
                                  "../../assets/Intake/intake.svg",
                                  height: 50,
                                  width: 50,
                                  color: Colors.black,
                                ), // SvgPicture.asset
                              ), // Container
                              Spacer(),
                              Container(
                                width: 200,
                                height: 25,
                                child: Expanded(child: Text("Inname")),
```

Figure 1 - Tiles

Image home page



Custom stepper

Our next step was to build a stepper, we did some research on flutter and found a build in stepper (Flutter, Flutter - Stepper class, sd), our internship supervisor told us to make a custom stepper. The reason for this was that the build-in stepper does not support functions we need and making an own stepper will help us to be a better programmer.

To develop a custom stepper, we needed to learn about state management. I spent a very long time in this part of programming as I found it quite difficult. You have stateless widget and stateful widget (Flutter, Flutter - StatefulWidget class, sd), stateless widget (Flutter, Flutter - Stateless Widget class, sd) you can't change the state and with stateful widget you can. I also had to apply this to my project which I found very difficult. it was only after a week or 2 with the help of my internship supervisor that I managed to do this.

Code

```dart
You, 3 weeks ago | 3 authors (You and others)
class Intake extends StatefulWidget {
  Intake({Key? key}) : super(key: key);

  @override
  State<Intake> createState() => _IntakeState();
}

You, 3 days ago | 2 authors (You and others)
class _IntakeState extends State<Intake> {
  String _callbackString = "";
  late CarType _carType;
  late List<bool> intake;
  late List<XFile> images;

  @override
  Widget build(BuildContext context) {
    print(context);
    // StepperViewModel vm = Provider.of<StepperViewModel>(context);
    return ChangeNotifierProvider(
        create: (_) => StepperViewModel(),
        child: Consumer2<StepperViewModel, IntakeViewModel>(
            builder: (_, foo, intake, __) {
          foo.addSteps({
            "VIN": ScanOrType(
              scannedCallback: ((p0) => {
                    _callbackString = p0,
                    setState(() {
                      _callbackString = p0;
                    }),
                    print("intake.dart " + _callbackString)
                  }),
            ), // ScanOrType
            "Info": CarTypeSelector(),
            "Check": Controle(),
            "Foto": fotoselector(),
          }, false);

          return StepperView();
        })); // Consumer2 // ChangeNotifierProvider
  }
}
```

Figure 5 - Intake.dart

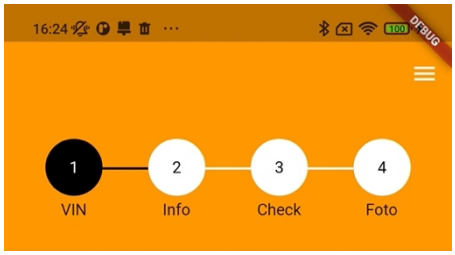
```dart
You, 4 weeks ago | 2 authors (You and others)
class StepperContainer extends StatelessWidget {
  const StepperContainer({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    StepperViewModel vm = Provider.of<StepperViewModel>(context);

    List<StepIndicator> steps = vm.steps.entries
        .mapIndexed((i, e) => StepIndicator(
              step: i,
              name: e.key,
            ))
        .toList();

    return ListView(
      shrinkWrap: true,
      scrollDirection: Axis.horizontal,
      children: [
        ...steps
            .mapIndexed((i, e) => Row(children: [
                  Padding(
                    padding: EdgeInsets.only(top: 12.0, bottom: 12.0),
                    child: e,
                  ), // Padding
                  i != steps.length - 1
                      ? Container(
                          padding: EdgeInsets.only(bottom: 18.0),
                          height: 100,
                          width: 40,
                          child: Divider(
                            color: vm.currentStep == i
                                ? ■Colors.black
                                : vm.currentStep > i
                                    ? ■Colors.green
                                    : □Colors.white,
                            thickness: 2,
                          ), // Divider
                        ) // Container
                      : Container(),
                ])) // Row
            .toList()
      ],
    ); // ListView
```

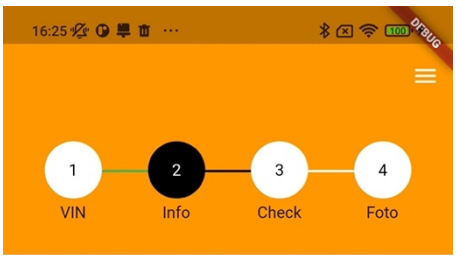
Figure 4 - Steppercontainer.dart

```dart
You, 4 weeks ago | 1 author (You)
class StepIndicator extends StatelessWidget {
  final int step;
  final String name;
  const StepIndicator({Key? key, required this.step, required this.name})
      : super(key: key);

  @override
  Widget build(BuildContext context) {
    StepperViewModel vm = Provider.of<StepperViewModel>(context);

    return GestureDetector(
      child: Container(
        child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [
          Container(
            width: 50,
            height: 50,
            decoration: BoxDecoration(
              color: vm.currentStep == step ? ■Colors.black : □Colors.white,
              borderRadius: BorderRadius.circular(50.0),
            ), // BoxDecoration
            child: Center(
                child: Text(
              (this.step + 1).toString(),
              style: TextStyle(
                  color: vm.currentStep == step ? □Colors.white : ■Colors.black), // TextStyle
            )), // Text // Center
          ), // Container
          Padding(padding: EdgeInsets.all(2.0)),
          Text(name)
        ]), // Column
      ), // Container
      onTap: () {
        vm.navigateTo(step);
      },
    ); // GestureDetector
  }
}
```

Figure 3 - stepperindicator.dart

Image





Step 1 – input vin

After we made the stepper, we started with the first step which was entering VIN. We used the QR-code/barcode scanner component from the old version of the app which worked fine. This step has also a page where you can type in VIN.

Code

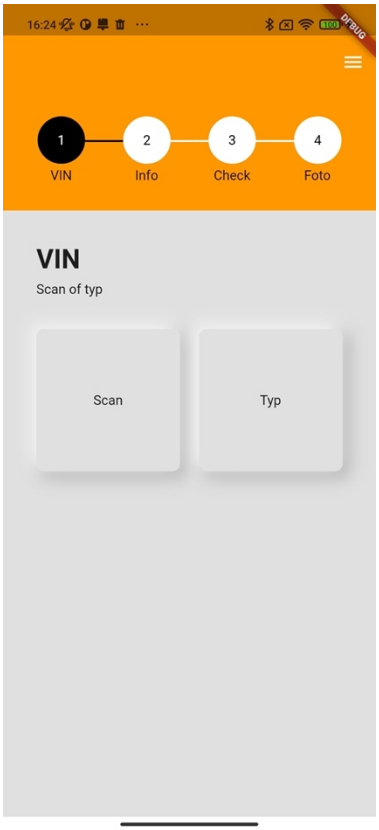
```dart
You, 3 weeks ago | 2 authors (You and others)
class ManualInput extends StatelessWidget {
  Function(String p1) inputCallback;

  ManualInput({Key? key, required this.inputCallback}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    IntakeViewModel Im = Provider.of<IntakeViewModel>(context);

    String localInput = "";
    Consumer<IntakeViewModel>(builder: (context, Im, child) {
      return Text('vin: ${Im.getVin}');
    }); // Consumer
    return Scaffold(
        resizeToAvoidBottomInset: false,
        appBar: defaultAppBar(),
        endDrawer: DrawerView(),
        backgroundColor: Colors.grey[300],
        body: Column(children: [
          Container(
            height: 150,
            width: double.infinity,
            color: Colors.orange,
          ), // Container
          Padding(padding: EdgeInsets.only(top: 10.0, bottom: 10.0)),
          Expanded(
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 8, vertical: 16),
                  child: Container(
                      child: Text(
                    "Vin",
                    style: TextStyle(
                        fontWeight: FontWeight.bold, fontSize: 28.0), // TextStyle
                  ))), // Text // Container // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 8, vertical: 16),
                  child: TextField(
                    onChanged: ((value) => {
                          localInput = value,
                        }),
                    maxLength: 17,
                    decoration: InputDecoration(
                      border: OutlineInputBorder(),
                      hintText: 'chassisnummer',
                    ), // InputDecoration
                  ), // TextField
                ), // Padding
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 8, vertical: 16),
                    child: Row(
                      mainAxisAlignment: MainAxisAlignment.end,
                      children: [
                        MaterialButton(
                            color: localInput.length == 17
                                ? Colors.orange
                                : Colors.orange,
                            onPressed: () {
                              inputCallback(localInput);
                              Im.updateVin(localInput.toString());      You, 3 weeks ago • Front-end info i
                              print(localInput.toString());

                              Navigator.pop(context);
                            },
                            child: Text("Ga verder ",
                                style: TextStyle(color: Colors.white))) // Text // MaterialButton
                      ],
                    )) // Row // Padding
              ],
            ), // Column
          ), // Expanded
          Container(
            height: 150,
            width: double.infinity,
            color: Colors.orange,
          ), // Container
        ])); // Column // Scaffold
  }
}
```

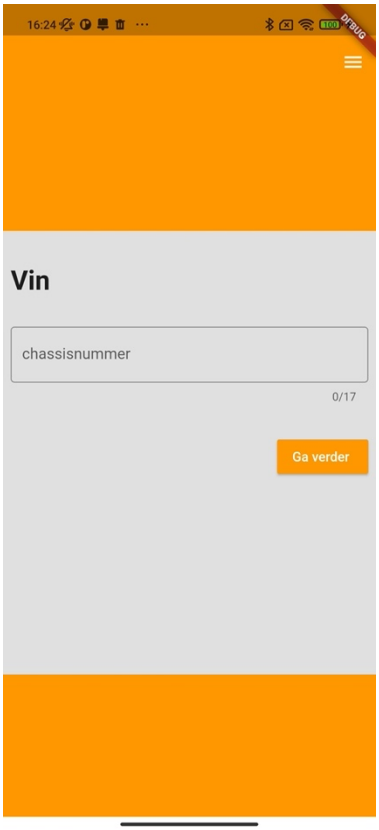
Figure 7 - Manualinput.dart

```dart
You, 3 weeks ago | 2 authors (You and others)
class ScanOrType extends StatefulWidget {
  Function(String p1) scannedCallback;

  ScanOrType({Key? key, required this.scannedCallback}) : super(key: key);
  String _callbackValue = "";

  @override
  State<ScanOrType> createState() => _ScanOrTypeState();
}

You, 3 weeks ago | 1 author (You)
class _ScanOrTypeState extends State<ScanOrType> {
  callback(String callbackValue) {
    setState(() {
      print(callbackValue);
      widget._callbackValue = callbackValue;
      widget.scannedCallback(callbackValue);
    });
  }

  @override
  Widget build(BuildContext context) {
    StepperViewModel vm = Provider.of<StepperViewModel>(context);
    IntakeViewModel im = Provider.of<IntakeViewModel>(context);

    return Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Row(
          children: [
            Container(
              height: 40,
              child: Text(
                "VIN ",
                style: TextStyle(fontWeight: FontWeight.bold, fontSize: 28.0),
              ), // Text
            ) // Container
          ], // Row
        ),
        Row(
          children: [
            Container(height: 50, width: 250, child: Text("Scan of typ")),
          ],
        ), // Row
        Row(children: [
          Expanded(
            child: GestureDetector(
              onTap: () {
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (BuildContext context) =>
                            ScannerCallbackComponent(
                                scannedCallback: (String callbackValue) => {
                                      im.updateVin(callbackValue.toString()),
                                      vm.nextStep()
                                    },
                                intentionText: "Scan",
                                doPop: true))); // ScannerCallbackComponent // M
              },
              child: tileButton("Scan"),
            ), // GestureDetector
          ), // Expanded
          Padding(padding: EdgeInsets.only(left: 10.0, right: 10.0)),
          Expanded(
            child: GestureDetector(
              onTap: () {
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (BuildContext context) => ManualInput(
                            inputCallback: (String callback) => {
                                  this.callback(callback),
                                  vm.nextStep(),
                                }))); // ManualInput // MaterialPageRoute
              },
              child: tileButton("Typ"),
            ), // GestureDetector
          ) // Expanded
        ]), // Row
      ],
    ); // Column
  }
}
```

Figure 6 - scanortype.dart

Image





Step 2 – Info

Soon we found out each step create its own routes, as result we could not pass the VIN to the next step. We wanted the VIN to be visible on each step. To solve this, we did some research on the web and found out about MVVM (Model view/view model) (Mohite, sd). MVVM is useful to move business logic from view to ViewModel and Model. ViewModel is the mediator between View and Model which carry all user events and return the result.

Code

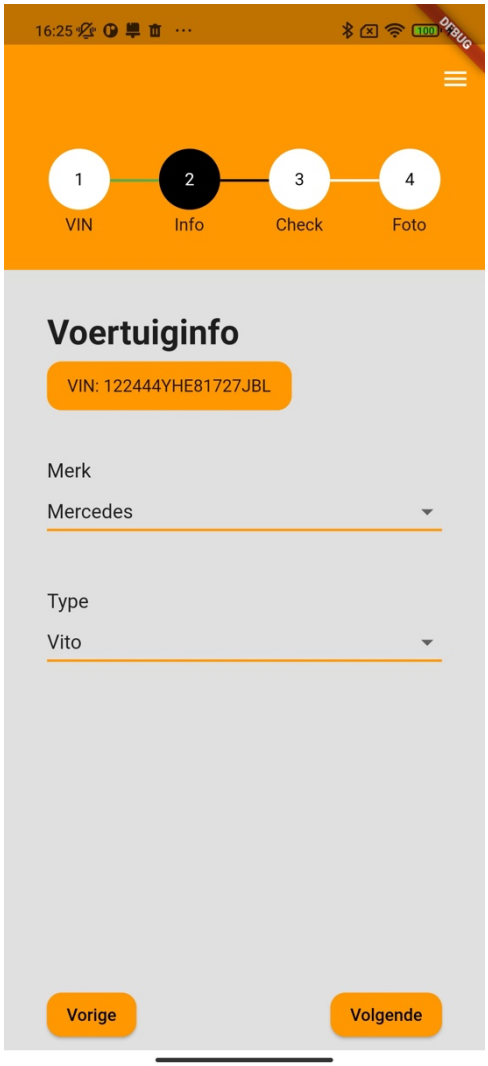
```
You, 3 days ago | 1 author (You)
class IntakeViewModel extends ChangeNotifier {
  String _vin = "";
  List<XFile> _files = [];

  void updateVin(String vin) {
    _vin = vin;
    notifyListeners();
  }      You, 3 weeks ago • worked on the IntakeViewM

  String get getVin {
    return _vin;
  }

  List<XFile> get files {
    return _files;
  }

  void addImage(XFile file) {
    _files.add(file);
    notifyListeners();
  }
}
```

Figure 9 - Intakeviewmodel.dart

```
@override
Widget build(BuildContext context) {
  IntakeViewModel im = Provider.of<IntakeViewModel>(context);      You, 3 week
  StepperViewModel vm = Provider.of<StepperViewModel>(context);

  return Container(
    child: Column(children: [
      Row(children: [
        Container(
          height: 40,
          child: Text(
            "Voertuiginfo",
            style: TextStyle(fontWeight: FontWeight.bold, fontSize: 28.0),
          ), // Text
        ) // Container
      ]), // Row
      Row(children: [
        Container(
            width: 200,
            height: 40,
            decoration: BoxDecoration(
                color: Colors.orange,
                borderRadius: BorderRadius.circular(12)), // BoxDecoration
            child: Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [Text("VIN: " + im.getVin)])), // Row // Container
      ]), // Row
```

Figure 8 - cartyeselector.dart

Image



Step 3 – Control

In this step we used the model view to get the VIN. Also, we needed to create a neomorphism button to do this we created a component. Furthermore, we added a text field so users can type in comments. On the bottom we added 2 buttons, one to go to previous step and one to go to the next step.

We wanted the button to be default on “Nee”, unfortunately I didn’t had time to program this. This still had to be done, this will improve the user experience. Now de buttons are not clickable. We still have to program a on press function for this button.

Code

```dart
Widget smallButton(buttoninput) {
  return Container(
    height: 50.0,
    width: 100,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(8.0),
      color: Colors.grey[300],      You, 3 weeks ago • Front-end info is do
      boxShadow: [
        BoxShadow(
          color: Colors.grey[500]!,
          blurRadius: 15,
          spreadRadius: 1,
          offset: const Offset(4.0, 4.0), // shadow direction: bottom right
        ), // BoxShadow
        BoxShadow(
          color: Colors.white,
          offset: Offset(-4.0, -4.0),
          blurRadius: 15,
          spreadRadius: 1,
        ), // BoxShadow
      ],
    ), // BoxDecoration
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisAlignment: MainAxisAlignment.center,
      children: [Text(buttoninput)],
    ), // Column
  ); // Container
}
```

Figure 10 - conponent small button

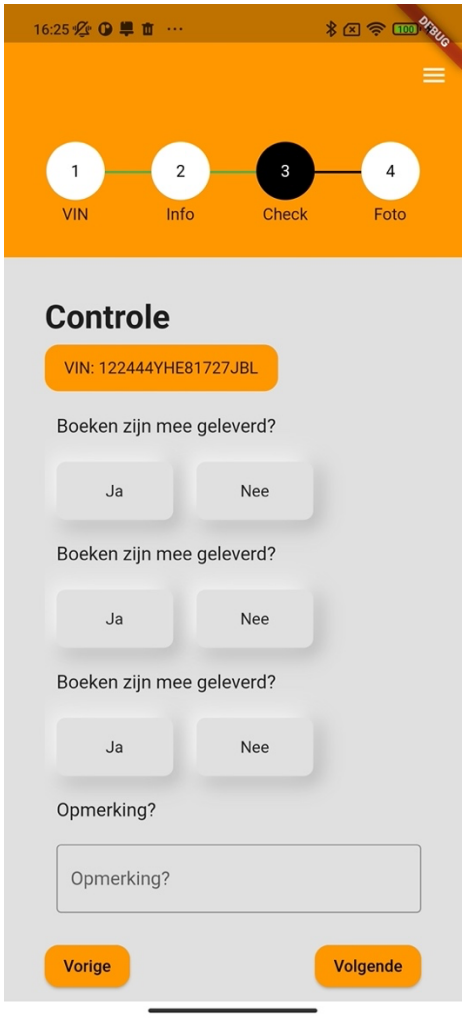
```dart
ElevatedButton(
  style: ButtonStyle(
    backgroundColor:
        MaterialStateProperty.all<Color>(Colors.orange),
    shape:
        MaterialStateProperty.all<RoundedRectangleBorder>(
      RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(12.0),
        side: BorderSide(
          color: Colors.orange,
          width: 2.0,
        ), // BorderSide
      ), // RoundedRectangleBorder
    ),
  ), // ButtonStyle
  child: Text(
    'Vorige',
    style: TextStyle(color: Colors.black),
  ), // Text
  onPressed: () {
    vm.previousStep();
  },
), // ElevatedButton
```

Figure 11 - control.dart > next step

```dart
class StepperViewModel extends ChangeNotifier {      You, 4 weeks ago • Fix
  int _currentStep = 0;

  String stepname = "";

  Map<String, Widget> _steps = {};

  int get currentStep => _currentStep;

  Map<String, Widget> get steps => _steps;

  Widget getCurrentStepWidget() => _steps.values.elementAt(currentStep);

  void navigateTo(int step) {
    _currentStep = step;
    notifyListeners();
  }

  void nextStep() {
    if (_currentStep + 1 == _steps.length) {
      return;
    }
    _currentStep++;
    notifyListeners();
  }

  void previousStep() {
    if (_currentStep == 0) {
      return;
    }
    _currentStep--;
    notifyListeners();
  }

  addSteps(Map<String, Widget> kv, [forceAdd = false]) {
    if (forceAdd) {
      _steps.addAll(kv);
      notifyListeners();
      return;
    }
    if (_steps.length > 0) {
      return;
    }
    _steps.addAll(kv);
  }
}
```

Figure 12 - stepper view model

Image



Step 4 – photo

This step users can take picture of the vehicle and add a comment. Because we used the existent project, we can reuse the camera widget. Furthermore, we created a component for the button.

Code

```
110         Padding(padding: EdgeInsets.only(top: 30.0)),
111         Row(
112           children: [
113             Container(height: 50, width: 250, child: Text("Opmerking")),
114           ],
115         ), // Row
116         TextField(
117           style: TextStyle(fontSize: 16.0, color: Colors.black),
118           decoration: InputDecoration(
119             border: OutlineInputBorder(),
120             labelText: '',
121           ), // InputDecoration
122         ), // TextField
```

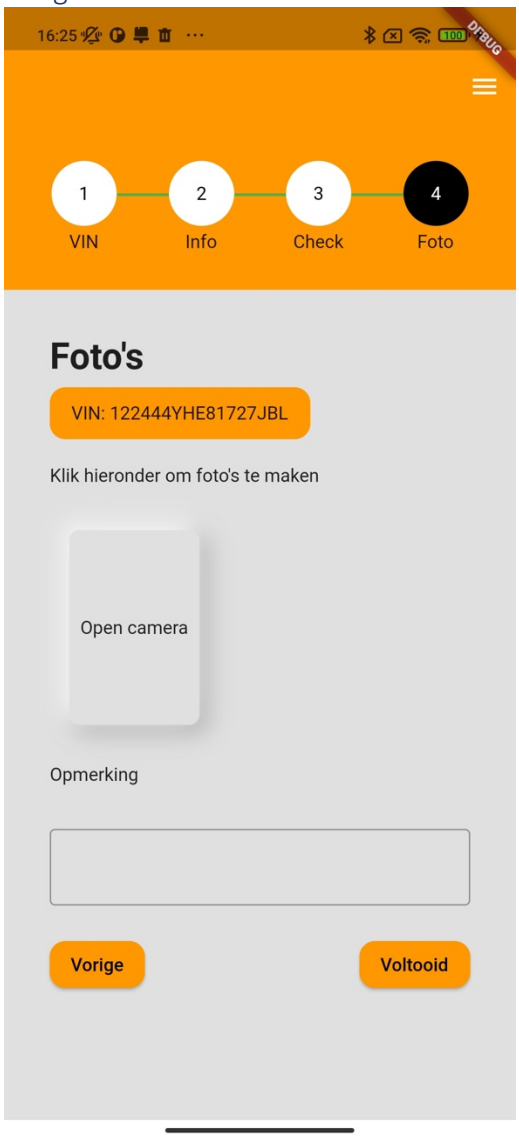
Figure 13 - fotoselector.dart > textfield

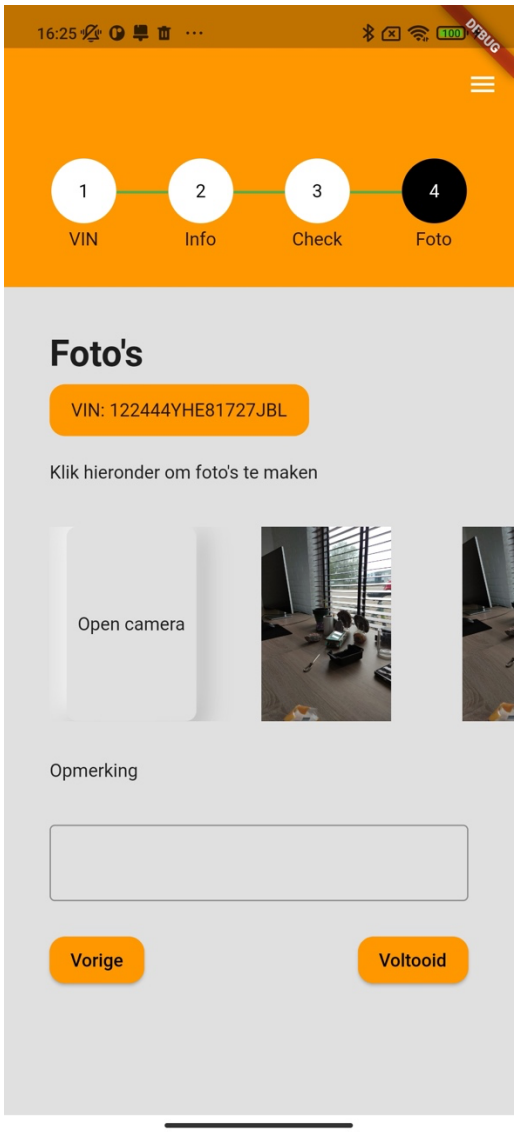
```
11  class fotoselector extends StatefulWidget {
12    const fotoselector({Key? key}) : super(key: key);
13
14    @override
15    State<fotoselector> createState() => _fotoselectorState();
16  }
17
    You, 5 seconds ago | 1 author (You)
18  class _fotoselectorState extends State<fotoselector> {
19    @override
20    Widget build(BuildContext context) {
21      IntakeViewModel im = Provider.of<IntakeViewModel>(context);
22      StepperViewModel vm = Provider.of<StepperViewModel>(context);
23
24      return Container(
25          child: Column(children: [
26        Row(children: [
27          Container(
28            height: 40,
29            child: Text(
30              "Foto's",
31              style: TextStyle(fontWeight: FontWeight.bold, fontSize: 28.0),
32            ), // Text
33          ) // Container
34        ]), // Row
35        Row(children: [
36          Container(
37              width: 200,
38              height: 40,
39              decoration: BoxDecoration(
40                  color: Colors.orange, borderRadius: BorderRadius.circular(12)),
41              child: Row(
42                  mainAxisAlignment: MainAxisAlignment.center,
43                  children: [Text("VIN: " + im.getVin)])), // Row // Container
44        ]), // Row
45        Padding(padding: EdgeInsets.only(top: 20.0)),
46        Row(
47          children: [
48            Container(
49                height: 50,
50                width: 250,
51                child: Text("Klik hieronder om foto's te maken")), // Container
52          ],
53        ), // Row
```

```
54        Container(
55          height: 150,
56          child: ListView(
57            scrollDirection: Axis.horizontal,
58            children: [
59              Row(
60                children: [
61                  Padding(padding: EdgeInsets.only(left: 5, right: 10)),
62                  tilerectangle("Open camera", context, im.addImage),
63                  Padding(padding: EdgeInsets.only(left: 10, right: 10)),
64                  new FutureBuilder<List<Image>>(
65                    future: Future.wait(im.files
66                        .map((e) async => Image.memory(await e.readAsBytes()))
67                        .toList()), // a Future<String> or null
68                    builder: (BuildContext context,
69                        AsyncSnapshot<List<Image>> snapshot) {
70                      print(snapshot.data!.length);
71                      switch (snapshot.connectionState) {
72                        case ConnectionState.none:
73                          return new Text('No image');
74                        case ConnectionState.waiting:
75                          return new Text('Awaiting result...');
76                        default:
77                          if (snapshot.hasError)
78                            return new Text('Error: ${snapshot.error}');
79                          else
80                            return Row(children: [
81                              ...snapshot.data!
82                                  .map((e) => Row(children: [
83                                        Padding(
84                                            padding: EdgeInsets.only(
85                                          left: 5,
86                                        )), // EdgeInsets.only // Padding
87                                        Container(
88                                          height: 150,
89                                          width: 150,
90                                          child: Ink.image(
91                                            image: e.image,
92                                            child: InkWell(
93                                              onTap: () {
94                                                print("test " + e.toString());
95                                              },
96                                            ), // InkWell
97                                          ), // Ink.image
98                                        ) // Container
99                                      ])) // Row
```

Figure 14 - fotoselector.dart > fotoselector

Image





Check part

Unfortunately, we didn't have the time to realize the check part but, this part is not so difficult. The front-end looks almost the same as the intake part. The check app has the type or scan VIN page and the photo selector page.

Sideproject

Our internship supervisor wanted to create a new website for Plexicastle. Plexicastle already had a WordPress website, but it was terrible. So, he decided he wanted to build a new blog-based website with next.js and tailwind. He chose to do this because it was just very easy to maintain.

We worked on this project for a week, we learned a lot about tailwind and next.js. Tailwind is very easy to use. I will defiantly use this often in future projects.

For this project we were not allowed to share the code. Instead, you can go to the plexicastle website https://www.plexicastle.com/ to see the result.

Portfolio

I made my portfolio from scratch, I used plain html, CSS and JavaScript. I made a one pager for my portfolio. With a vertical navbar on the left. I added few animations when scrolling and the navbar stays static when you scroll down.

Github link: https://github.com/thupten1/Portfolio_2021

Jira

We also used Jira as a scrum tool for our portfolio, we used this to create tasks and for the planning. With Jira our internship coordinator could keep up with our work. Furthermore, this gave us a view of what needed to do.

Tasks on Jira for portfolio

STAGT-27	Make leeswijzer	Unassigned	thupten rekonkati	=	DONE	Unresolved	30 May 2022	30 May 2022		
STAGT-26	learn about View Model	Unassigned	thupten rekonkati	=	DONE	Unresolved	30 May 2022	30 May 2022		
STAGT-25	stage management	Unassigned	thupten rekonkati	=	DONE	Done	14 Apr 2022	30 May 2022		
STAGT-24	learn about widgets and classes	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	8 Apr 2022		
STAGT-23	learn about flutter	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	14 Apr 2022		
STAGT-22	start on portfolio	Unassigned	thupten rekonkati	=	IN PROGRESS	Unresolved	8 Apr 2022	8 Apr 2022		
STAGT-21	learn about the structure of flutter	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	8 Apr 2022		
STAGT-20	learn about global theme	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	8 Apr 2022		
STAGT-19	start with learning flutter	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	14 Apr 2022		
STAGT-18	make a design using design principle and keep and research insights	Unassigned	thupten rekonkati	=	DONE	Unresolved	8 Apr 2022	14 Apr 2022		
STAGT-17	Gather the research inputs and prioritise them	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	14 Apr 2022		
STAGT-16	create a document	Unassigned	thupten rekonkati	=	TO DO	Unresolved	8 Apr 2022	8 Apr 2022		
STAGT-15	study the outcome	Unassigned	thupten rekonkati	=	DONE	Unresolved	8 Apr 2022	8 Apr 2022		
STAGT-14	Take survey	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	8 Apr 2022		
STAGT-13	Make survey	Unassigned	thupten rekonkati	=	DONE	Done	8 Apr 2022	8 Apr 2022		
STAGT-12	Interview-knowledge-mobile-technology	Unassigned	thupten rekonkati	=	IN PROGRESS	Unresolved	21 Mar 2022	14 Apr 2022		
STAGT-11	Empathy map	Unassigned	thupten rekonkati	=	DONE	Done	21 Mar 2022	21 Mar 2022		
STAGT-10	Interview wishes and needs	Unassigned	thupten rekonkati	=	DONE	Done	9 Mar 2022	23 Mar 2022		
STAGT-9	Interview Wessel	Unassigned	thupten rekonkati	=	DONE	Done	21 Feb 2022	24 Feb 2022		
STAGT-7	Research the target audience	Unassigned	thupten rekonkati	=	DONE	Done	21 Feb 2022	8 Apr 2022		
STAGT-6	Research current situation	Unassigned	thupten rekonkati	=	DONE	Done	21 Feb 2022	9 Mar 2022		
STAGT-5	Create Approach and Planning	Unassigned	thupten rekonkati	=	DONE	Done	8 Feb 2022	21 Feb 2022		
STAGT-4	Write project assignment	Unassigned	thupten rekonkati	=	DONE	Done	8 Feb 2022	21 Feb 2022		
STAGT-3	Get more insight about the current situation	Unassigned	thupten rekonkati	=	DONE	Done	8 Feb 2022	21 Feb 2022		
STAGT-2	Introduction	Unassigned	thupten rekonkati	=	DONE	Done	8 Feb 2022	21 Feb 2022		
STAGT-1	Create a project plan	Unassigned	thupten rekonkati	=	DONE	Done	8 Feb 2022	22 Feb 2022	19 F	

List of Figures

Bibliography

Mohite, J. (n.d.). *Flutter: MVVM Architecture*. From Medium: https://medium.com/flutterworld/flutter-mvvm-architecture-f8bed2521958

Marshall, M. (n.d.). *flutter/website*. From GitHub: https://github.com/flutter/website/blob/main/src/development/ui/layout/index.md

Flutter. (n.d.). *Flutter - Stepper class*. From Flutter docs: https://api.flutter.dev/flutter/material/Stepper-class.html#:~:text=A%20material%20stepper%20widget%20that,widget%20is%20a%20flexible%20wrapper.

Flutter. (n.d.). *Flutter - StatefulWidget class*. From Flutter docs: https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html

Flutter. (n.d.). *Flutter - Stateless Widget class*. From Flutter docs: https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html